# Table of Contents



# Assignment Multiple regression

This assignment is of an applied nature and uses data that are available in the data file TestExer2-GPA-round2. The question of interest is whether the study results of students in Economics can be predicted from the scores on entrance tests taken before they start their studies. More precisely, you are asked to investigate whether verbal and mathematical entrance tests predict freshman grades of students in Economics.

## A

1. Regress FGPA on a constant and SATV. Report the coefficient of SATV and its standard error and p-value (give your answers with 3 decimals).
2. Determine a 95% confidence interval (with 3 decimals) for the effect on FGPA of an increase by 1 point in SATV.

## B

Answer questions $a_i$ $a_i$ and $a_2$ $a_2$ also for the regression of FGPA on a constant, SATV, SATM, and FEM.

## C

Determine the (4 × 4) correlation matrix of FGPA, SATV, SATM, and FEM. Use these correlations to explain the differences between the outcomes in parts (A) and (B).

## D

1. Perform an F-test on the significance (at the 5% level) of the effect of SATV on FGPA, based on the regression in part (b) and another regression. Note: Use the F-test in terms of SSR or $R^2$ $R^2$ and use 6 decimals in your computations. The relevant critical value is 3.9.
2. Check numerically that $F = t^2$ $F = t^2$.

### TestExer2-GPA Data

- Data on student learning of 609 students
- FGPA: Freshman grade point average (scale 0-4)
- SATV: Score on SAT Verbal test (scale 0-10)
- SATM: Score on SAT Mathematics test (scale 0-10)
- FEM: Gender dummy (1 for females, 0 for males)

# Import data & Libraries

In [9]:

```
import numpy as np
import pandas as pd

from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import f_regression as fp
```

In [15]:

```
TestExer = pd.read_csv('/users/downloads/TestExer2-GPA-round2.csv', sep = ','
TestExer.head()
```

Out[15]:

| | Observation | FGPA | SATM | SATV | FEM |
|---|---|---|---|---|---|
| **0** | 1 | 2.52 | 4.0 | 4.0 | 1 |
| **1** | 2 | 2.33 | 4.9 | 3.1 | 0 |
| **2** | 3 | 3.00 | 4.4 | 4.0 | 1 |
| **3** | 4 | 2.11 | 4.9 | 3.9 | 0 |
| **4** | 5 | 2.15 | 4.3 | 4.7 | 0 |

# Part A code

In [16]:

```
SATV = TestExer.SATV.values.reshape(-1,1) # independent variable
FGPA = TestExer.FGPA # dependent variable
```

In [17]:

```
model_a = LinearRegression()
model_a.fit(X=SATV, y=FGPA)
print("The coefficient rounded to 3 decimals is: ", round(model_a.coef_[0], 3
print("The intercept rounded to 3 decimals is: ", round(model_a.intercept_, 3
```

```
The coefficient rounded to 3 decimals is:  0.063
The intercept rounded to 3 decimals is:  2.443
```

In [18]:

```
F_test, P_value = fp(SATV, FGPA)
print("The P-Value rounded to 3 decimals is: ", round(P_value[0], 3))
```

```
The P-Value rounded to 3 decimals is:  0.023
```

In [19]:

```
FGPA_pred = model_a.predict(SATV)
SSE_a = ((FGPA - FGPA_pred)**2).sum() # sum of squared error
s_a = np.sqrt((SSE_a)/(len(FGPA)-2)) # standard error for part a
s_b_sq = s_a**2 / ((SATV - SATV.mean())**2).sum() # std. error sq of slope
s_b = np.sqrt(s_b_sq)
```

In [20]:

```
lower_limit = (model_a.coef_ - 1.96*s_b)
upper_limit = model_a.coef_ + 1.96*s_b
conf_interval = [lower_limit[0], upper_limit[0]]
```

## Summary Part A

In [21]:

```
# Part a - 1: coefficient of SATV and its standard error and p-value
print("The coefficient rounded to 3 decimals is: ", round(model_a.coef_[0], 3
print("The Standard error of Slope rounded to 3 decimals is: ", round(s_b, 3)
print("The P-Value rounded to 3 decimals is: ", round(P_value[0], 3))

# Part a - 2: 95% confidence interval (with 3 decimals)
print("\n")
print("The 95% confidence interval for effect on FGPA with an increase by 1 p
```

```
The coefficient rounded to 3 decimals is:  0.063
The Standard error of Slope rounded to 3 decimals is:  0.028
The P-Value rounded to 3 decimals is:  0.023


The 95% confidence interval for effect on FGPA with an increase by 1
point is:
 [0.008732742834394507, 0.11720204595554712]
```

## Part B code

In [22]:

```
X = TestExer[["SATV", "SATM", "FEM"]] # independent variable
y = TestExer.FGPA # dependent variable
```

In [23]:

```
model_b = LinearRegression()
model_b.fit(X, y)
np.around(model_b.coef_, 3)
```

Out[23]:

```
array([0.014, 0.173, 0.2  ])
```

In [24]:

```
y_pred = model_b.predict(X)
SSE_a2 = ((y - y_pred)**2).sum() # sum of squared error
s_a2 = np.sqrt((SSE_a2)/(len(X)-2)) # standard error
s_b2_sq = s_a2**2 / ((TestExer.SATV - TestExer.SATV.mean())**2).sum() # SSE S
s_b2 = np.sqrt(s_b2_sq)
s_b2
```

Out[24]:

```
0.026609143982420952
```

In [25]:

```
lower_limit = (model_b.coef_[0] - 1.96*s_b2)
upper_limit = model_b.coef_[0] + 1.96*s_b2
conf_interval = [lower_limit, upper_limit]
conf_interval
```

Out[25]:

```
[-0.038207815448865784, 0.06610002896222435]
```

## Summary Part B

In [26]:

```
# Part a - 1: coefficient of SATV and its standard error
print("The coefficient rounded to 3 decimals is: ", round(model_b.coef_[0], 3
print("The Standard error of Slope rounded to 3 decimals is: ", round(s_b2, 3

# Part a - 2: 95% confidence interval (with 3 decimals)
print("\n")
print("The 95% confidence interval for effect on FGPA with an increase by 1 p
```

```
The coefficient rounded to 3 decimals is:  0.014
The Standard error of Slope rounded to 3 decimals is:  0.027


The 95% confidence interval for effect on FGPA with an increase by 1
point is:
 [-0.038207815448865784, 0.06610002896222435]
```

## Part C Code

In [27]:

```
subset_df = TestExer[["FGPA", "SATV", "SATM", "FEM"]]
subset_df.corr()
```

Out[27]:

| | FGPA | SATV | SATM | FEM |
|---|---|---|---|---|
| **FGPA** | 1.000000 | 0.091972 | 0.195378 | 0.176428 |
| **SATV** | 0.091972 | 1.000000 | 0.287801 | 0.033577 |
| **SATM** | 0.195378 | 0.287801 | 1.000000 | -0.162680 |
| **FEM** | 0.176428 | 0.033577 | -0.162680 | 1.000000 |

## Summary Part C

In general, SATV has significant impact on FGPA when it was the only feature. However, since SATM and SATV are the correlated, the total significance comes from SATM influence. When there was a partial dependence (Case B), we saw that SATV does not have a significant impact. Now FEM and SATV do not have a significant correlation so it mut be maintained in the model. The effect of SATM can be absorbed by SATV.

## Part D Code

Suing the results of Part B and inferences of Part C, we can create a new model which looks at only SATM and FEM. We can use that to determine SSR which is defined as the sum of the squared differences between the prediction for each observation and the population. We cab then use the results fo this model to compare against the model_b.

In [28]:

```
model_h0 = LinearRegression()
X_h0 = TestExer[["SATM", "FEM"]]
y_h0 = TestExer.FGPA
```

In [29]:

```
model_h0.fit(X_h0, y_h0)
y_pred_h0 = model_h0.predict(X_h0)

SSR_h0_sq = (y_h0 - y_pred_h0)**2
SSR_h0 = SSR_h0_sq.sum()
r_sq_h0 = r2_score(y_h0, y_pred_h0)
```

In [30]:

```
SSR_b_sq = (y - y_pred)**2
SSR_b = SSR_b_sq.sum()
r_sq_b = r2_score(y, y_pred)
```

In [31]:

```
F = (SSR_h0 - SSR_b) / (SSR_b / 605)
```

## Part D Summary

In [32]:

```
# Modified Model - SSR and R-squared
print("The Modified Model SSR rounded to 6 decimals is: ", format(SSR_h0, '.6
print("The Modified Model R^2 rounded to 6 decimals is: ", format(r_sq_h0, '.

# Part B Model - SSR and R-squared
print("\n")
print("The Part-B Model SSR rounded to 6 decimals is: ", format(SSR_b, '.6f')
print("The Part-B Model R^2 rounded to 6 decimals is: ", format(r_sq_b, '.6f'

# F Statistic
print("\n")
print("The F statistic rounded to 3 decimals is: ", format(F, '.3f'))
```

```
The Modified Model SSR rounded to 6 decimals is:  118.191220
The Modified Model R^2 rounded to 6 decimals is:  0.082703


The Part-B Model SSR rounded to 6 decimals is:  118.142539
The Part-B Model R^2 rounded to 6 decimals is:  0.083081


The F statistic rounded to 3 decimals is:  0.249
```

Since the value of the F-statistic is less than the provided critical value, we can safely conclude that the Null hypothesis $H_0 H_0$ is not rejected.

In [ ]:

```
 
```